Latent Semantic Indexing Based on Factor Analysis

Noriaki Kawamae

Center for Advanced Research and Technology The University of Tokyo
4-6-1 Komaba Meguroku Tokyo153-8904 JAPAN
tel. +81- 3-5452-5277 fax. +81- 5452-5278
kawamae@mpeg.rcast.u-tokyo.ac.jp

Abstract

The main purpose of this paper is to propose a novel latent semantic indexing (LSI), statistical approach to simultaneously mapping documents and terms into a latent semantic space. This approach can index documents more effectively than the vector space model (VSM). Latent semantic indexing (LSI), which is based on singular value decomposition (SVD), and probabilistic latent semantic indexing (PLSI) have already been proposed to overcome problems in document indexing, but critical problems remain. In contrast to LSI and PLSI, our method uses a more meaningful, robust statistical model based on factor analysis and information theory. As a result, this model can solve the remaining critical problems in LSI and PLSI. Experimental results with a test collection showed that our method is superior to LSI and PLSI from the viewpoints of information retrieval and classification. We also propose a new term weighting method based on entropy.

1. Introduction

With the advent of digital databases and communication networks, it is easy to obtain Web pages and electronic documents. It is thus necessary to develop information retrieval methods to simplify the process of finding relevant information. Document indexing is one important information retrieval method. Traditionally, documents have been indexed and labeled manually by humans. An important example is the idea of notional families in the work of H. P. Luhn [9]. The primary goal is to index documents with the same precision achieved by humans. To develop such document indexing methods, the following problems must be solved:

◍Ambivalence between terms
◍Calculation cost
◍Document indexing and keyword matching methods

Due to these problems, the retrieval performance of indexing systems is poor. Among previous works, latent semantic indexing (LSI), based on singular value decomposition (SVD), and probabilistic latent semantic indexing (PLSI) have been developed to overcome these problems, but unsolved problems remain.

Our primary goal in this paper is to present a novel statistical approach, to simultaneously mapping documents and terms into a latent semantic space. This approach can index documents better than by using individual indexing terms because the topics in a document are more closely related to the concepts described therein than to the index terms used in the document's description. Our method uses a more meaningful, robust statistical model - called a code model - that associates documents and terms with a latent semantic factor. This model is based on factor analysis and information theory and enables us to remove this noise and extract a better latent semantic space than other methods. As a result, documents can be indexed nearly as well as by humans. This is mainly because factor analysis is a better statistical model than SVD for capturing hidden factors.

2. Related work on document indexing and term selection

The vector space model (VSM) [11] is an approach to mapping documents into a space associated with the terms in the documents. The weighting of the terms in a document provides the document's coordinates in space, and the similarity between documents is measured in space.

Latent semantic analysis (LSA) [2] is an approach to mapping documents into a lower dimensional space than in LSI. This is accomplished by projecting term vectors into this space by using a model based on SVD. To date, several theoretical results or explanations have appeared, and these studies have provided a better understanding of LSI. However, many fundamental problems remain unresolved, as follows.

Inadequate use of statistical methods:
LSI uses SVD as a statistical model. Therefore, LSI cannot explain or extract the latent semantic factor as a hidden variable, because SVD is nothing more than the decomposition of the observed variable matrix [6].

Optimal decomposition dimension:
In general, dimensionality reduction is justified by the statistical significance of the latent semantic vectors as measured by the likelihood of the model based on SVD [3],[5]. The complexity of the model is not considered in terms of the dimension.

Term weighting method:
To date, many researchers on LSI have used tf/idf [11] or a similar method. This type of method, however, is not the best way to evaluate the usefulness of terms, because the weighting of low-frequency terms is underestimated, while that of high-frequency terms is overestimated.

3. Latent semantic extraction and term selection

Our method is a novel statistical approach to simultaneously mapping documents and terms into a latent semantic space, and it improves document retrieval performance. Our method consists of three components: (1) a novel term weighting method, (2) a code model, and (3) a statistical model criterion. The main idea in this approach is that a latent semantic factor is associated with each topic. A particular topic in a document is more related to the concepts described in the document than to the index terms used in the description of the document.

Therefore, this proposed indexing method enables us to retrieve documents based on similarities between concepts.

As a result, our proposed method evolves from keyword matching to concept matching. This allows us to retrieve documents even if they are not indexed by the terms in a query, because one document shares concepts with another document indexed by the given query. Therefore, the latent semantic space improves document retrieval performance.

3.1 Term-document matrix

Morphological analysis can be used to convert a document into a vector consisting of the terms occurring in it. The vector space model is a method of geometrically visualizing the relationships between documents. In this model, the relationships between terms and documents are represented as a term-document matrix, which contains the values of the index terms *t* occurring in each document *d*, properly weighted by other factors [4][12][8]. We denote *m* as the number of index terms in a collection of documents and *n* as the total number of documents. Formally, we let *A* denote a term-document matrix with n rows and m columns and let w_{ij} be an element (i, j) of *A*. Each w_{ij} is assigned a weight associated with the term-document pair (d_i, t_j), where d_i $(1 \leq i \leq n)$ represents the *i*-th document and t_j $(1 \leq j \leq m)$ represents the *j*-th term. For example, using a *tf/idf* representation, we have $w_{ij} = tf(t_i\text{-}d_i)idf(t_j)$. Thus, given the values of the w_{ij}, the term-document matrix represents the whole document collection. Therefore, each document can be expressed as a vector consisting of the weights of each term and mapped in a vector space:

$$A = \begin{pmatrix} w_{11} & \bullet & \bullet & \bullet & w_{1n} \\ \bullet & \bullet & & & \bullet \\ \bullet & & \bullet & & \bullet \\ \bullet & & & \bullet & \bullet \\ w_{m1} & \bullet & \bullet & \bullet & w_{mn} \end{pmatrix} \equiv \begin{pmatrix} d_1 & \bullet & \bullet & \bullet & d_n \end{pmatrix} \equiv \begin{pmatrix} t_1 \\ \bullet \\ \bullet \\ \bullet \\ t_m \end{pmatrix}. \qquad (1)$$

There are two methods of term weighting [8]: local and global. Among various weighting methods, those designated as L3, G2, and G3 are novel and have not been described previously [Toku 1999].

3.1.1 Local weighting

This approach defines the weights w_{ij} in each document. Let P_{ij} $(1 \leq i \leq n, 1 \leq j \leq m)$ denote the occurrence probability of t_j in d_i. We ascribe significance to a term's occurrence, on the grounds that it represents a document's topics more than other factors do. Therefore, we base w_{ij} on the term occurrence probability P_{ij} in each document, and we define a local weighting L_{ij} as follows:

$$L_{ij} = P_{ij} * log\ (1+P_{ij}). \qquad (2)$$

In contrast to other local weighting methods based on term frequency, this method reduces the effect of very high frequency terms by multiplying P_{ij} by a logarithm.

3.1.2 Global weighting

This approach defines the weights w_{ij} over all documents. Let P_{ij} $(1 \leq i \leq n,\ 1 \leq j \leq m)$ denote the relative occurrence probability of t_j in d_j. We ascribe significance to the probability of a term occurring over all documents on the grounds that a given term provides information for topic prediction and affects global performance. Entropy is a convenient metric for comparing the probability distribution of a term's occurrence. Therefore, we base w_{ij} on the entropy of the relative occurrence probability, and we define a global weighting G_j as follows:

$$G_j = 1 + \frac{1}{\log n} \sum_{i=1}^{n} p_{ij} \log p_{ij}\ . \qquad (3)$$

Because this weighting is based on entropy, if a term occurs with equal probability in all documents, it is weighted as 1 (the maximum). To ensure that it does not exceed 1, this weighting is normalized through division by *log n*, where *n* is the total number of documents.

3.2 Introduction of factor analysis to obtain latent semantic space

3.2.1 Code model

The main theme in obtaining a latent semantic space is to capture the essential, meaningful semantic associations while reducing the amount of redundant, noisy information. Here, we propose a code model to determine a document's coordinates in the latent semantic space. The code model is based on the hypothesis that terms in documents are generated from a latent semantic factor and a given document has a probability of belonging to some category. This model can be used to determine the coordinates of not only documents but also terms. The relationships between terms and latent semantic factors are similar to the relationship between encoded data and an information source in information theory. Because of this similarity, we call our model a code model.

We next describe the key idea in this model. We think that a latent semantic factor is an information source and is coded into a term in a document. Therefore, the relationship between terms and latent semantic factors can be defined as follows.

$P(t_j|l_k)$: Probability of latent semantic factor l_k generating term t_j,

where $\sum_{j=i}^{m} P(t_j|l_k) = 1$. A term t_j can be generated from not only the latent semantic factor l_k but also another factor $l_{k'}$.

The relationship between a document and a latent semantic factor can be defined in the following way:

$P(l_k|d_i)$: Probability of document d_i belonging to latent semantic factor l_k, where $\sum_{i=1}^{n} P(l_k|d_i)=1$. Document d_i can belong to not only latent semantic factor l_k but also another factor $l_{k'}$.

The weights w_{ij}, which represent the value of t_j in d_i, are used to combine these definitions into a joint probability model, resulting in the expression:

$$w_{ij} = P(t_j|d_i) = \sum_{k=1}^{l} P(t_j|l_k)P(l_k|d_i) + P_\varepsilon(t_j)P_\varepsilon(d_i), \qquad (4)$$

where $P(t_j|d_i)$ denotes not the empirical occurrence but the statistical probability distribution of t_j in d_i, obtained by multiplying the local and global weightings; $P_\varepsilon(t_j)$ denotes the unique probability of t_j; and $P_\varepsilon(d_i)$ denotes the unique probability of d_i. The reason for introducing these quantities is that $P(t_j|d_i)$ cannot actually be explained in terms of only the joint probability of $P(t_j|l_k)$ and $P(l_k|d_i)$. The difference between the statistical probability distribution $P(t_j|d_i)$ and the model distribution based on only the joint probability of $P(t_j|l_k)$ and $P(l_k|d_i)$ is considered as noise in information theory. Therefore, we call this model a code model.

Notice that multiple documents can belong to some latent semantic factor at the same time. This is because the latent semantic factors are associated with the observed variances as t_j, d_i. Moreover, the latent semantic factors do not constrain documents to be orthogonal to each other.

Despite the similarity, the fundamental difference between the aspect model [5] in PLSI and this code model is the unique probability. This difference affects the latent semantic factors obtained.

3.2.3 Factor analysis

To calculate the occurrence probability of terms in the code model, we introduce factor analysis, which is a statistical method that resolves an observed variance into a corresponding latent factor. Undoubtedly, SVD can also calculate a variable corresponding to the latent semantic factor obtained arithmetically by our method. The variable in SVD is a mixture of observed variables. In contrast, the variable in factor analysis is a latent factor. Therefore, it fits the code model mentioned earlier. In utilizing factor analysis, we define the observed variance as $P(t_j|d_i)$ and the latent factor as a concept.

3.3 Statistical model selection based on stochastic complexity (SC)

A term-document matrix is composed of only observed variances. The object of factor analysis is to estimate the factor loading, meaning $P(l|d)$ in the code model, while keeping the number of unique factors as low as possible. We need to define the optimized number of factors in advance. In this paper, we determine this number by applying stochastic complexity (SC) [Rissanen 1996] one time. This refers to a code length that shortest in the case of coding with the number of parameters fixed to m. SC is defined as follows to determine the optimized number of latent semantic factors:

$$SC(A|k) \cong -\log P_{\hat{a}}(A) + \frac{[2n(k+1)-k(k-1)]}{2}\log(n*m \qquad (10)$$

where $P_a(A)$ denotes the maximum likelihood estimator of term-document matrix A. The minimum of SC($A|k$) gives the optimized number of latent semantic factors, k. This formulation allows us to solve the remaining problem of determining the optimized number of factors in both factor analysis and (P)LSI.

The code model not only determines documents coordinates in a latent semantic space but also is a proper generative model for the observed data, i.e., the probability of t_j in d_i.

4. Document indexing in latent semantic space

4.1 Experimental design

Here, we assess the effectiveness of OUR METHOD from three viewpoints: (1) term weighting methods, (2) Our method vs. (P)LSI, and (3) statistical model selection based on SC.

Both (P)LSI and OUR METHOD use the term-document matrix as a starting point and map documents into a latent semantic space. To compare the effectiveness of these methods, we evaluated the retrieval performance for latent semantic spaces based on each method. Or, in terms of obtaining a latent semantic space, we compared SVD and factor analysis. We evaluated term weighting in terms of the contribution to the latent semantic factors, and statistical model selection in terms of whether it can predict the optimal number of latent semantic factors.

For our experiments, we prepared 210 news articles to form a test collection. These articles covered seven topic categories: economics, entertainment, information technology, politics, society, sports, and world news. Each category included the same number of articles. We used these categories to judge retrieval and classification results in terms of average precision and recall rate.

First, we decomposed the articles into terms by using morphological analysis, which exchanges a sentence for its component terms. We used Chasen's approach [1] for this step. We used only terms that meet the condition that the part of speech is a noun or an unknown that is not registered in the dictionary of morphological analysis. This is because the terms, as used in queries, are limited to these parts of speech. We also omitted terms that did no (7) ar in three or less different documents. Generally, terms with a size of 1 (for example, “a”, &, @, etc.) were defined as stop words and omitted, and numbers were omitted as well. Single Japanese characters, however, do have meaning, so we did not treat these as stop words. The documents contain 7863 different terms, so a 7863 * 210 term-document matrix was generated.

4.2 Comparison of (P)LSI vs Our method and term weighting methods

Table1 compares similarity results among the documents based on different term weightings and latent semantic spaces. This comparison was done as follows. First, we calculated the documents' similarity matrix, as defined in section 3.3.5 Second, if the similarity of two documents was above a threshold, we judged these documents to be alike. Even if one of the documents did not include a term used in a query, we included both documents in the search results. Finally, we evaluated the search results in terms of the precision and recall rates, as shown in Table 1. In this experiment, we defined 0.7 as the threshold.

The normal vector space model consisted of 7863 term axes. The spaces decomposed by SVD or by factor analysis both consisted of 7 axes. In both cases, the precision and recall rate were highest for a decomposed space with this number of axes.

Table 1 indicates that L3 and L4 were the most effective term weighting methods for local weighting, while G1, G3, and G4 were the most effective for global weighting. As for the decomposed space, both SVD and factor analysis achieved higher values than the normal vector space model. These results show that the similarity between documents measured in a space decomposed by SVD or factor analysis reflects a fundamental relationship between topics in the documents. They do not, however, distinguish Our method from (P)LSI. The term weighting methods used in this experiment are defined as follows.

Local weighting

L1: term occurrence probability (=tf)

Weights w_{ij} defined by occurrence in a document.

$$L1_{ij} = P_{ij} = \frac{C(t_{ij})}{\sum_{j=1}^{m_j} C(t_{ij})}$$

L2: normalized term occurrence probability

Weights w_{ij} defined by a normalization of L1 based on a log function.

$$L2_{ij} = \log\left(1 + \frac{C(t_{ij})}{\sum_{j=1}^{m_j} C(t_{ij})}\right)$$

L3: normalized entropy of document

Weights d_j defined by a term's occurrence probability distribution in a document.

$$L3_i = 1 - \frac{1}{\log m_i}\sum_{j=1}^{m_i} p_{ij} \log p_{ij}$$

m_i: Number of different kinds of terms in document d_i.
P_{ij}: Frequency probability of term t_j in document d_i.

L4: proposed local weighting method

Global weighting

G1: normalized entropy of term occurrence over all documents

Weights t_j defined by each term's occurrence over all documents. This method is similar to our proposed global weighting but differs in terms of P_{ij}.

$$G_j = 1 + \frac{1}{\log n}\sum_{i=1}^{n} p_{ij} \log p_{ij}$$

P_{ij}: Occurrence probability of term t_{ij} over all documents.

G2: normalized entropy of term occurrence over all documents

Weights t_j defined by the occurrence proportion over all documents.

$$G2_j = -p_j \log p_j - (1 - p_j)\log(1 - p_j)$$

P_j: Occurrence probability of a document containing t_j over all documents.

G3: proposed global weighting method

G4: document frequency (=idf)

Weights d_i defined by the inverse of the document frequency.

$$H_j = 1 + \log\frac{n}{C(d_j)}$$

$C(t_j)$: Number of documents containing t_i.

G5: normalized entropy of relative term occurrence

Weights d_j defined by the relative probability distribution over all documents.

$$H_j = 1 + \frac{1}{\log n}\sum_{i=1}^{n} p_{ij} \log p_{ij}$$

P_{ij}: Relative probability of t_j occurring in d_i.

Table 1.: Average precision and recall rates in for (P)LSI and Our method

This table compares not only the term weighting methods but also theVSM, (P)LSI, and Our method, in for each term weighting method. In For each same term weighting method, the upper top row is the VSM, the middle row is (P)LSI, and the lower bottom row is Our method. Data listed as N/P/R. TW and N/P/R are defined bellow this table.

TW	Economics	eEntertainment	ITinformation	Politics	sSports	sSocietyal	World news	aAverage
L1G1	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.7/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	28.7/96.6/92.7	30.0/98.9/98.7	30.0/98.7/98.7	30.0/99.7/99.6	30.1/99.8/100	30.3/98.3/99.1	29.1/98.7/95.8	29.7/98.7/97.8
	28.7/96.3/92.4	30.0/98.9/98.9	30.2/98.6/99.1	30.0/100/99.8	29.9/99.8/99.6	30.3/98.1/99.1	29.0/98.7/95.6	29.8/98.6/97.8

L1G2	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.9/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	28.3/97.0/91.8	29.7/100/99.1	29.5/98.7/97.1	30.0/100/99.8	29.9/100/99.8	29.6/99.4/98.2	20.1/98.5/66.2	28.2/99.1/93.1
	27.8/96.7/90.0	29.7/100/99.1	29.8/98.8/98.0	29.9/100/99.8	29.7/100/98.9	29.1/99.4/96.7	18.9/98.6/62.2	27.8/99.1/92.1
L1G3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.8/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	28.9/96.9/93.6	30.0/99.3/99.3	30.1/98.9/99.3	29.9/99.8/99.6	30.1/99.8/100	30.1/98.8/99.1	29.1/98.7/95.8	29.8/98.9/98.1
	28.8/96.5/92.9	30.1/99.4/99.8	30.2/98.7/99.3	30.0/100/99.8	30.1/99.8/100	30.1/98.8/99.1	29.0/98.7/95.6	29.8/98.8/98.1
L1G4	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.8/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	29.6/97.8/96.4	30.1/99.7/100	30.2/99.0/99.6	30.1/99.8/100	30.1/99.7/100	30.0/99.5/99.6	30.0/99.2/99.3	30.0/99.2/99.3
	29.7/97.5/96.4	30.2/99.5/100	30.2/98.8/99.6	30.1/100/100	30.0/99.5/99.6	30.0/99.5/99.3	30.1/99.1/99.3	30.0/99.1/99.2
L2G1	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.7/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	28.7/96.6/92.7	29.9/99.0/98.7	30.0/98.7/98.7	30.0/99.7/99.6	30.1/99.8/100	30.2/98.4/99.1	29.1/98.7/95.8	29.7/98.7/97.8
	28.7/96.3/92.4	30.0/99.0/98.9	30.2/98.6/99.1	30.0/100/99.8	29.9/99.8/99.6	30.3/98.2/99.1	29.0/98.7/95.6	29.7/98.6/97.8
L2G2	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.9/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	28.3/97.0/91.8	29.7/100/99.1	29.6/98.8/97.6	30.0/100/99.8	29.9/100/99.8	29.6/99.4/98.2	20.3/98.6/66.9	28.2/99.1/93.3
	27.8/96.8/90.2	29.7/100/99.1	29.8/98.8/98.2	29.9/100/99.8	29.7/100/98.9	29.1/99.4/96.7	18.9/98.6/62.2	27.9/99.1/92.2
L2G3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.8/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	29.0/97.0/94.0	30.0/99.4/99.3	30.1/98.9/99.3	29.9/99.8/99.6	30.1/99.8/100	30.1/98.9/99.1	29.1/98.7/95.8	29.8/98.9/98.2
	28.9/96.5/93.1	30.1/99.4/99.8	30.2/98.7/99.3	30.0/100/99.8	30.1/99.8/100	30.1/98.8/99.1	29.0/98.7/95.6	29.8/98.8/98.1
L2G4	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/99.8/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	29.6/97.8/96.4	30.1/99.7/100	30.2/99.0/99.6	30.1/99.8/100	30.1/99.7/100	30.0/99.5/99.6	30.0/99.2/99.3	30.0/99.2/99.3
	29.6/97.6/96.4	30.2/99.5/100	30.2/98.8/99.6	30.1/100/100	30.2/99.5/100	30.0/99.5/99.3	30.0/99.2/99.3	30.0/99.1/99.2
L3G1	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
L3G2	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/99.9/99.8	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100	29.1/100/97.1	29.9/100/99.6
	29.9/99.9/99.6	30.2/99.2/100	30.0/99.9/100	30.0/100/100	28.8/99.0/95.3	30.0/100/100	17.7/100/58.9	28.1/99.7/93.4
L3G3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
L3G4	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
L4G1	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
L4G2	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/99.9/99.8	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100	29.1/100/97.1	29.9/100/99.6
	29.9/99.9/99.6	30.2/99.2/100	30.0/99.9/100	30.0/100/100	28.8/99.0/95.3	30.0/100/100	17.7/100/58.9	28.1/99.7/93.4
L4G3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100
L4G4	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.0/100/3.3	1.1/100/3.6	1.0/100/3.4
	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100

	30.0/100/100	30.0/99.9/100	30.0/100/100	30.00.0/100	30.0/99.9/100	30.0/100/100	30.0/100/100	30.0/100/100

TW:term weighting, N:the number of documents retirievedaldocuments, P:precision, R:recall

documents, we cannot distinguish factor analysis from SVD in terms of precision. When we reduce the number of terms by using the value of global weighting, however, we see the superiority of factor analysis.

As for the weighting methods, the results indicate that it is effective to select terms based on the value with L4 as the local weighting method. We should emphasize that the latent semantic space based on the combination of factor analysis and G3*L4 can be used to classify documents with only the half number of terms normally required. The VSM, on the other hand, does not depend on the number of terms or the weighting method. hese results show that our proposed weighting method is useful not only for indexing documents but also for selecting the minimum number of terms.

4.4 Statistical model selection

In this experiment, the optimal number of latent semantic factors was seven. This is the same as the number of topics used. This indicates that the obtained latent semantic factors can effectively characterize documents statistically; we can say that these factors represent semantic meanings. SC predicts nine as the optimal number of latent semantic factors, as shown in Table 2. Other objective functions [3], [5], however, predict numbers far greater than/less than nine, which do not appear in Table 2. The value predicted by SC is thus approximately correct.

Table 2. Optimal number of latent semantic factors and SC

FN	5	6	7*	8	9
Likelihood	16924.56	16074.36	15217.97	14463.32	13784.99
SC	19529.03	19329.83	19124.53	19020.98	18993.73*

FN: Number of latent semantic factors

5. Conclusions

We have proposed a novel statistical approach to simultaneously mapping documents and terms into a latent semantic space. Our method consists of three components: (1) a novel term weighting method, (2) a code model, and (3) a statistical model criterion.

Retrieval and classification experiments on a test collection indicated that Our method is superior to (P)LSI. In other words, the axes in a latent semantic space obtained by our method are closer to the general concepts indexed by humans than any other method.

Finally, we introduced a statistical model selection approach based on stochastic complexity (SC) to solve the remaining problem in (P)LSI: the problem of how to determine the number of latent semantic factors. Our experiments showed that the formulation based on SC solves this problem.

We thus conclude that our method is a useful document indexing method that not only solves the critical remaining problems in LSI and PLSI but also improves the retrieval performance.

REFERENCES

[1] Chasen: http://chasen.aistnara.ac.jp/index.html.ja.

[2] Deerwester, S., Dumais, S. T., Furnas, G. W., Landaure, T. K., and Harshman, R.: Indexing by latent semantics analysis, Journal of the American Society for Information Science, 1990.

[3] Ding, C. H. Q.: A Dual Probabilistic Model for Latent Semantic Indexing in Information Retrieval and Filtering, in Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR), 1999.

[4] Dumais, S.T.: Improving the retrieval of information from external sources, Behavior Research Methods, Instruments and Computers, 23(2), 229-236. 1991.

[5] Hofmann, T.: Probabilistic latent semantic indexing, in Proceedings of the 22nd Annual Conference on Research and Development in Information Retrieval (ACM SIGIR), 1999.

[6] Kawamae, N., Aoki, T., Yasuda, H.: Information Retrieval Based on the Information Theory Model, Technical Report of IEICE, DE2001-57, 2001 (in Japanese).

[7] Kawamae, N., Aoki, T., Yasuda, H.: Document Classification and Retrieval after Removing Word Noise, Technical Report of IEICE, NLC2001-48, 2001 (in Japanese).

[8] Kita, K.: Statistical Language Model, The University of Tokyo Press, 1999.

[9] Luhn, H. P.: The automatic derivation of information retrieval encodement from machine readable text, Information Retrieval and Machine Translation, 3(2), 1021-1028, 1961.

[10] Schutze, H. and Pedersen, J.: A vector model for syntagmatic and paradigmatic relatedness, in Proceedings of the 9th Annual Conference of the University of Waterloo Center for the New OED and Text Research, 1993.

[11] Salton, G. and McGill, M. J.: Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[12] Salton, G. and Buckley, C.: Term-weighting approaches in automatic text retrieval, Information Processing and Management, 24(5) 513-523, 1988.

[13] Saul, L., and Pereira, F.: Aggregate and mixed order Markov models for statistical language processing, in Proceedings of 2nd International Conference on Empirical Methods in Natural Language Processing, 1997.

[14] Tohku, T.: Information Retrieval and Language Process, The University of Tokyo Press, 1999.